Kapitel ML: II

II. Grundlagen des Maschinellen Lernens

- Daten
- Datenexploration
- Konzeptlernen: Suche im Hypothesenraum
- Konzeptlernen: Suche im Version-Space
- Performance Measures

Daten

Skalen

Charakterisierung der Objekte in O durch Attributausprägungen. Mögliche Skalen für Attribute:

- Nominalskalen: $=$ und $\neq$
- Ordinalskalen: $=$, $\neq$ und $<$
- Intervallskalen: $=$, $\neq$, $<$ und feste Einheiten, also $-$
- Ratioskalen (Verhältnisskalen): reelle Zahlen, alle Operationen

Attributwerte sind

- diskret (Nominalskalen, Ordinalskalen) oder
- stetig (Intervallskalen, Ratioskalen).

Welche Skalen können von welchen Lernverfahren bearbeitet werden?

Bemerkungen:

- Als Werte von intervall- oder ratioskalierten Attributen verwenden wir nur reelle Zahlen.
- Auch Werte ordinal- oder nominalskalierter Attribute können durch reelle Zahlen codiert werden.
- Nominalskalierte Attribute heißen auch kategorisch; intervall- oder ratioskalierten Attribute heißen auch numerisch.

Daten

Probleme bei der Behandlung von Datenmengen

- Vielzahl von Datentypen:
  Die verschiedenen Skalen erlauben unterschiedliche Verarbeitungsmethoden ihrer Werte.

- Nonstandard-Daten:
  Die Vektoren zur Beschreibung der Objekte haben unterschiedliche Struktur und/oder Länge.

- Inhomogenität:
  Unterschiedliche Zusammenhänge zwischen Variablen gelten in verschiedenen Teilen des Instanzenraum.

- Dimensionalität:
  Die benötigte Anzahl von Datenpunkten für eine bestimmte Dichte wächst exponentiell mit der Dimension des Datenraumes (Curse of Dimensionality).

Kapitel ML: II (Fortsetzung)

II. Grundlagen des Maschinellen Lernens

- Daten
- Datenexploration
- Konzeptlernen: Suche im Hypothesenraum
- Konzeptlernen: Suche im Version-Space
- Performance Measures

Konzeptlernen: Suche im Hypothesenraum

Lernaufgabe

Gegeben sei eine Menge D von Beispielen: Tage, die mit den 6 Attributen „Sky“, „Temperature“, „Humidity“, „Wind“, „Water“ und „Forecast“ beschrieben sind – zusammen mit einer Aussage, ob gerne Sport betrieben wird.

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
1	sunny	warm	normal	strong	warm	same	yes
2	sunny	warm	high	strong	warm	same	yes
3	rainy	cold	high	strong	warm	change	no
4	sunny	warm	high	strong	cool	change	yes

- Was ist das Konzept hinter „EnjoySport“ ? – bzw.
- Was sind mögliche Hypothesen h, mit denen sich das Konzept / die Kategorie / die Klasse c „EnjoySport“ beschreiben lässt?

Bemerkungen:

- Wertebereiche der Merkmale in der Lernaufgabe:

Sky	**Temperature**	**Humidity**	**Wind**	**Water**	**Forecast**
sunny	warm	normal	strong	warm	same
rainy	cold	high	weak	cool	change
cloudy					

Konzeptlernen: Suche im Hypothesenraum

Lernaufgabe (Fortsetzung)

Aufbau einer Hypothese h:

1. Konjunktion von Constraints in Form von Attribut-Wert-Paaren
2. drei Arten von Werten: Literal, beliebig „?“, nichts erlaubt „$\perp$“

Beispiel für eine Hypothese: $\langle$ *sunny, ?, ?, strong, ?, same* $\rangle$

Konzeptlernen: Suche im Hypothesenraum

Lernaufgabe (Fortsetzung)

Aufbau einer Hypothese h:

1. Konjunktion von Constraints in Form von Attribut-Wert-Paaren
2. drei Arten von Werten: Literal, beliebig „?“, nichts erlaubt „$\perp$“

Beispiel für eine Hypothese: $\langle$ *sunny, ?, ?, strong, ?, same* $\rangle$

Definition 1 (Hypothese-erfüllend)
Ein Vektor $\mathbf{x} \in X$ erfüllt eine Hypothese h, in Zeichen $h(\mathbf{x}) = 1$, genau dann, wenn $\mathbf{x}$ mit allen von h vorgeschriebenen Literalen übereinstimmt und h kein „$\perp$“ enthält. Erfüllt $\mathbf{x}$ die Hypothese h nicht, gilt $h(\mathbf{x}) = 0$.

- die speziellste Hypothese: $s_0 = \langle \perp, \perp, \perp, \perp, \perp, \perp \rangle$
- die allgemeinste Hypothese: $g_0 = \langle$ *?, ?, ?, ?, ?, ?* $\rangle$

Konzeptlernen: Suche im Hypothesenraum

Lernaufgabe (Fortsetzung)

Das zu lernende Konzept c wird Zielkonzept, Zielfunktion oder Klassifikator genannt und bildet von der Menge der Beispiele nach $\{0, 1\}$ ab:

$$c : X \rightarrow \{0, 1\}, \qquad \text{z.B. } \textit{EnjoySport} : X \rightarrow \{0, 1\}$$

Beispiele mit $c(\mathbf{x}) = 1$ heißen positive, mit $c(\mathbf{x}) = 0$ negative Beispiele.

Eine `Trainingsmenge` D enthält positive und negative Beispiele für die Zielfunktion:

$$D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\}$$

Konzeptlernen: Suche im Hypothesenraum

Lernaufgabe (Fortsetzung)

Das zu lernende Konzept c wird Zielkonzept, Zielfunktion oder Klassifikator genannt und bildet von der Menge der Beispiele nach $\{0, 1\}$ ab:

$$c : X \to \{0, 1\}, \qquad \text{z.B. } \textit{EnjoySport} : X \to \{0, 1\}$$

Beispiele mit $c(\mathbf{x}) = 1$ heißen positive, mit $c(\mathbf{x}) = 0$ negative Beispiele.

Eine `Trainingsmenge` D enthält positive und negative Beispiele für die Zielfunktion:

$$D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\}$$

Definition 2 (Konsistenz)
Eine Hypothese h ist konsistent hinsichtlich eines Beispiels $(\mathbf{x}, c(\mathbf{x}))$, genau dann, wenn $h(\mathbf{x}) = c(\mathbf{x})$ gilt. h ist konsistent hinsichtlich einer Trainingsmenge D, in Zeichen *consistent*(h, D), genau dann, wenn gilt:

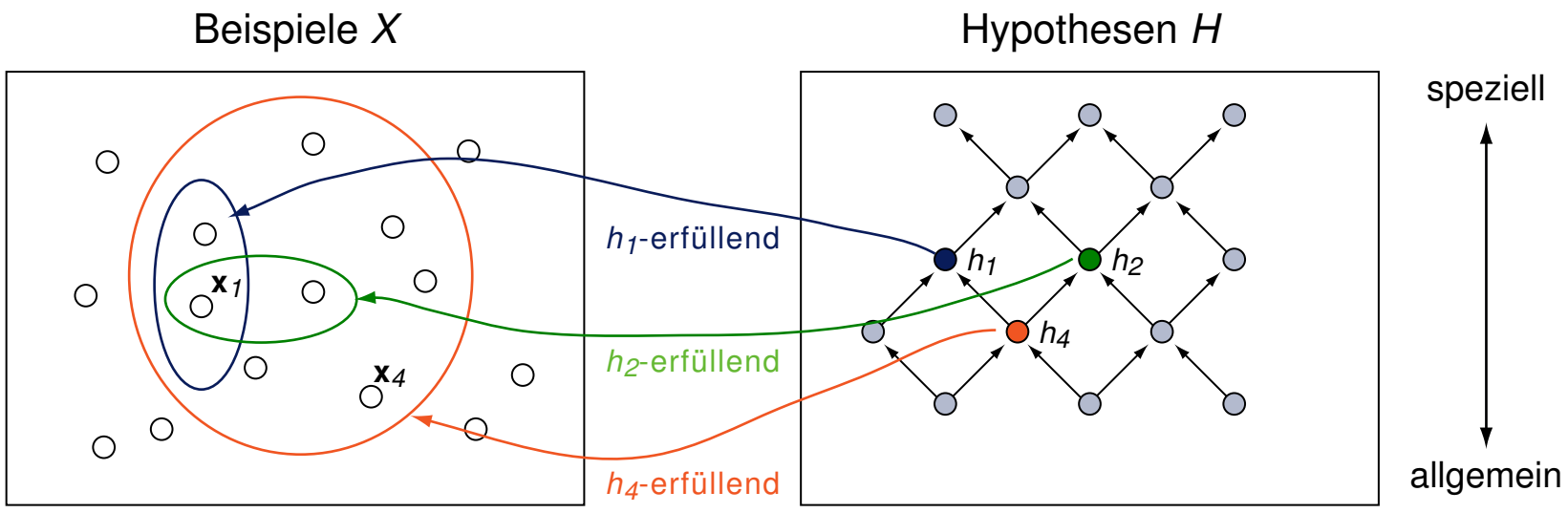
$$\forall(\mathbf{x}, c(\mathbf{x})) \in D : \; h(\mathbf{x}) = c(\mathbf{x})$$

Bemerkungen:

- Beachte für ein Beispiel $\mathbf{x}$ den Unterschied zwischen „eine Hypothese erfüllen“ und „mit einer Hypothese konsistent sein“. Im ersten Fall ist $h(\mathbf{x}) = 1$ gefordert; das Zielkonzept $c(\mathbf{x})$ bleibt unberücksichtigt. Im zweiten Fall ist die Übereinstimmung von Zielkonzept $c(\mathbf{x})$ und Hypothesenwert $h(\mathbf{x})$ im positiven wie im negativen Fall gefordert.
- Die Konsistenz einer Hypothese h kann bzgl. eines Beispiels – aber auch bzgl. einer Menge D von Beispielen betrachtet werden. Im letzteren Fall wird für alle $\mathbf{x}$ in D verlangt, dass $h(\mathbf{x}) = 1$ genau dann, wenn $c(\mathbf{x}) = 1$. Das ist äquivalent damit, dass $h(\mathbf{x}) = 0$ genau dann, wenn $c(\mathbf{x}) = 0$.
- Die Lernaufgabe besteht in der Bestimmung einer zu D konsistenten Hypothese h aus dem Hypothesenraum H.

Konzeptlernen: Suche im Hypothesenraum

Ordnung auf einem Hypothesenraum



$\mathbf{x}_1 = (\textit{sunny, warm, normal, strong, warm, same})$

$\mathbf{x}_4 = (\textit{sunny, warm, high, strong, cool, change})$

$h_1 = \langle\, \textit{sunny, ?, normal, ?, ?, ?} \,\rangle$

$h_2 = \langle\, \textit{sunny, ?, ?, ?, warm, ?} \,\rangle$

$h_4 = \langle\, \textit{sunny, ?, ?, ?, ?, ?} \,\rangle$

Konzeptlernen: Suche im Hypothesenraum

Ordnung auf einem Hypothesenraum

Definition 3 (generellere Hypothese)
Sei X eine Menge von Beispielen und seien h_1 und h_2 Funktionen mit Definitionsbereich X und Wertebereich $\{0, 1\}$. Dann heißt h_1 genereller-oder-gleich hinsichtlich h_2, in Zeichen $h_1 \geq_g h_2$, genau dann, wenn gilt:

$$\forall \mathbf{x} \in X : \; h_2(\mathbf{x}) = 1 \;\text{ impliziert }\; h_1(\mathbf{x}) = 1$$

h_1 ist streng genereller als h_2, in Zeichen $h_1 >_g h_2$, genau dann, wenn gilt:

$$(h_1 \geq_g h_2) \;\wedge\; (h_2 \not\geq_g h_1)$$

Bemerkungen:

- Wenn h_1 genereller als h_2 ist, so kann man umgekehrt auch davon sprechen, dass h_2 spezieller als h_1 ist.
- $\geq_g$ bzw. $>_g$ sind unabhängig von einem Zielkonzept c; sie hängen nur davon ab, ob die Beispiele die Hypothesen erfüllen, also, ob $h(\mathbf{x}) = 1$ ist – und nicht, ob $c(\mathbf{x}) = 1$ ist.
- Die $\geq_g$-Relation definiert eine partielle Ordnung auf dem Hypothesenraum H. Das heißt, die Relation ist reflexiv, antisymmetrisch und transitiv. Es handelt es um eine *partiell* – im Gegensatz zur einer *total* – geordneten Struktur, weil nicht alle Paare von Hypothesen in der Relation stehen: es gibt Hypothesen h_i, h_j, für die weder $h_i \geq_g h_j$ noch $h_j \geq_g h_i$ gilt.
- Über die Semantik von „impliziert“: „A impliziert B.“ $\sim$ „Wenn A dann B.“ $\sim$ „A erfordert B.“ $\sim$ „B geht mit A einher“. Insbesondere steht „impliziert“ *nicht* für „folgt“. Für eine Folgerung existiert eine Herleitung, also ein Beweis; aber aus $h_2(\mathbf{x}) = 1$ kann $h_1(\mathbf{x}) = 1$ nicht hergeleitet bzw. nicht bewiesen werden. Mit der Implikation wird lediglich eine Bedingung angegeben, die erfüllt sein muss, um unter die Definition zu fallen.

Konzeptlernen: Suche im Hypothesenraum

Induktive Lernannahme *(Inductive Learning Hypothesis)*

> *"Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples."*
>
> [p.23, Mitchell 1997]

Konzeptlernen: Suche im Hypothesenraum

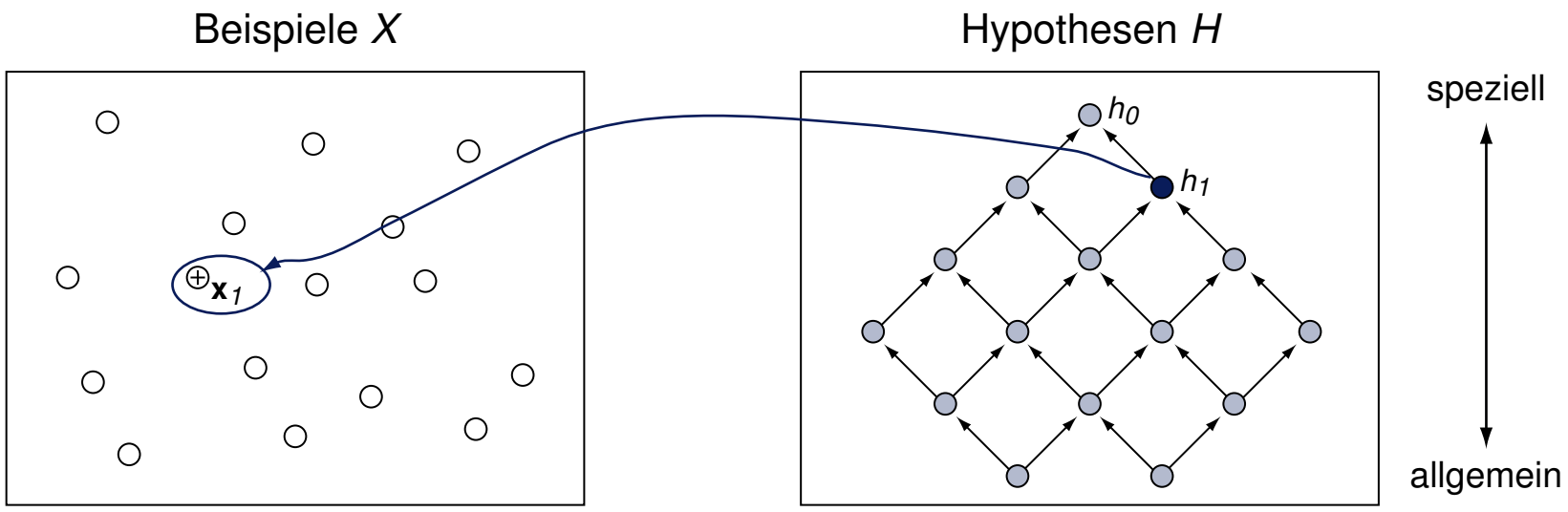
Algorithmus Find-S

1. $h = s_0$ // s_0 is the maximally specific hypothesis in H.
2. **FOREACH** $(\mathbf{x}, c(\mathbf{x})) \in D$ **DO**
   IF $c(\mathbf{x}) = 1$ **THEN** // Use only positive examples.
   FOREACH $a_i \in h$ **DO**
   UNLESS *satisfied*$(\mathbf{x}, a_i)$ **DO**
   relax(h, a_i) // Replace a_i by next more general constraint.
   ENDDO
   ENDDO
   ENDIF
   ENDDO
3. *return*(h)

Siehe Trainingsmenge D zum Konzept *EnjoySport*.

Konzeptlernen: Suche im Hypothesenraum

Algorithmus Find-S

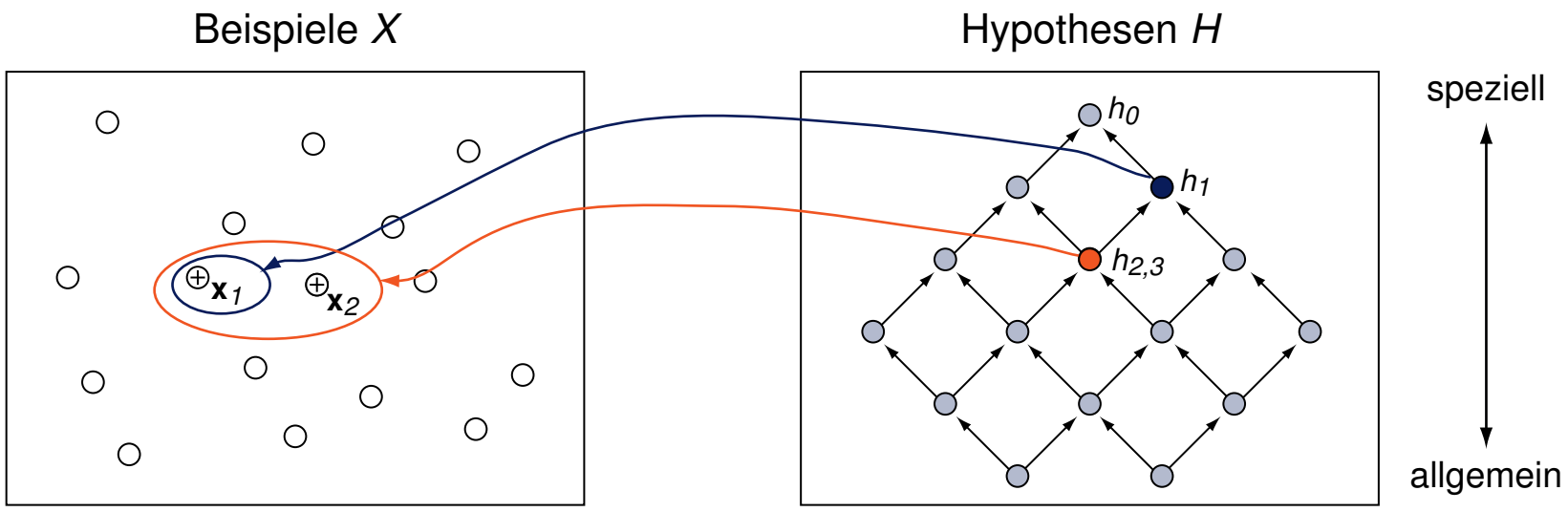


$h_0 = \langle \perp, \perp, \perp, \perp, \perp, \perp \rangle$

$\mathbf{x}_1 = (\textit{sunny, warm, normal, strong, warm, same})$ $h_1 = \langle \textit{sunny, warm, normal, strong, warm, same} \rangle$

Konzeptlernen: Suche im Hypothesenraum

Algorithmus Find-S



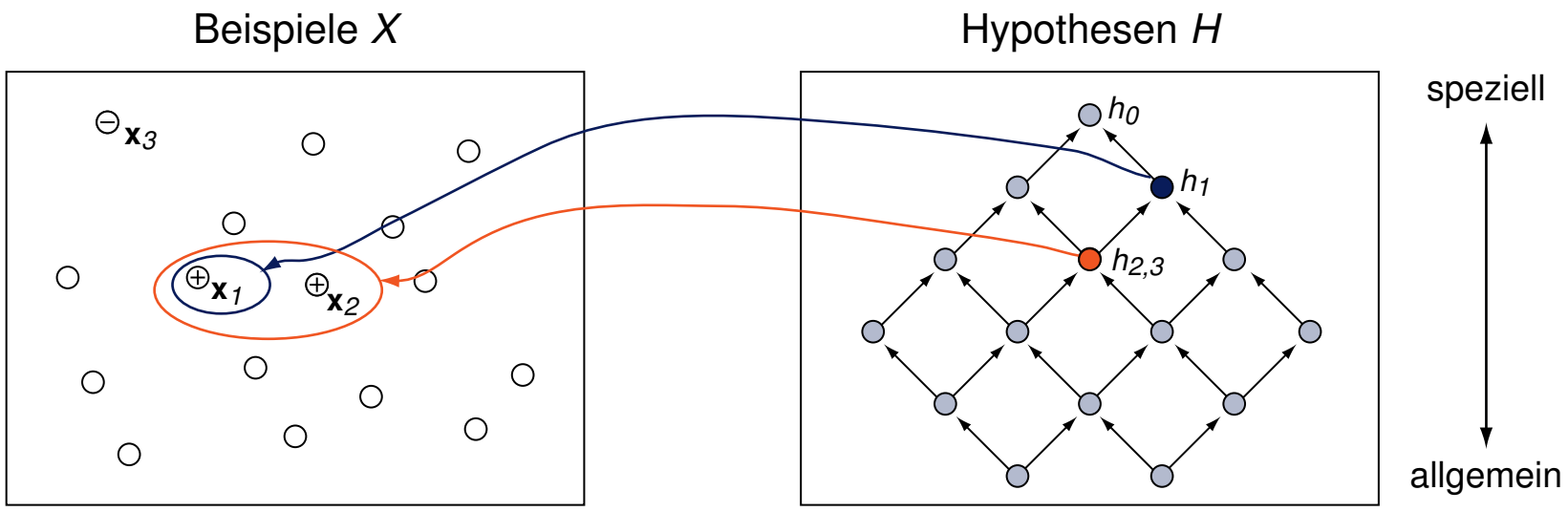
$h_0 = \langle \perp, \perp, \perp, \perp, \perp, \perp \rangle$

$\mathbf{x}_1 = (\textit{sunny, warm, normal, strong, warm, same})$ $h_1 = \langle \textit{sunny, warm, normal, strong, warm, same} \rangle$

$\mathbf{x}_2 = (\textit{sunny, warm, high, strong, warm, same})$ $h_2 = \langle \textit{sunny, warm, ?, strong, warm, same} \rangle$

Konzeptlernen: Suche im Hypothesenraum

Algorithmus Find-S



$h_0 = \langle \perp, \perp, \perp, \perp, \perp, \perp \rangle$

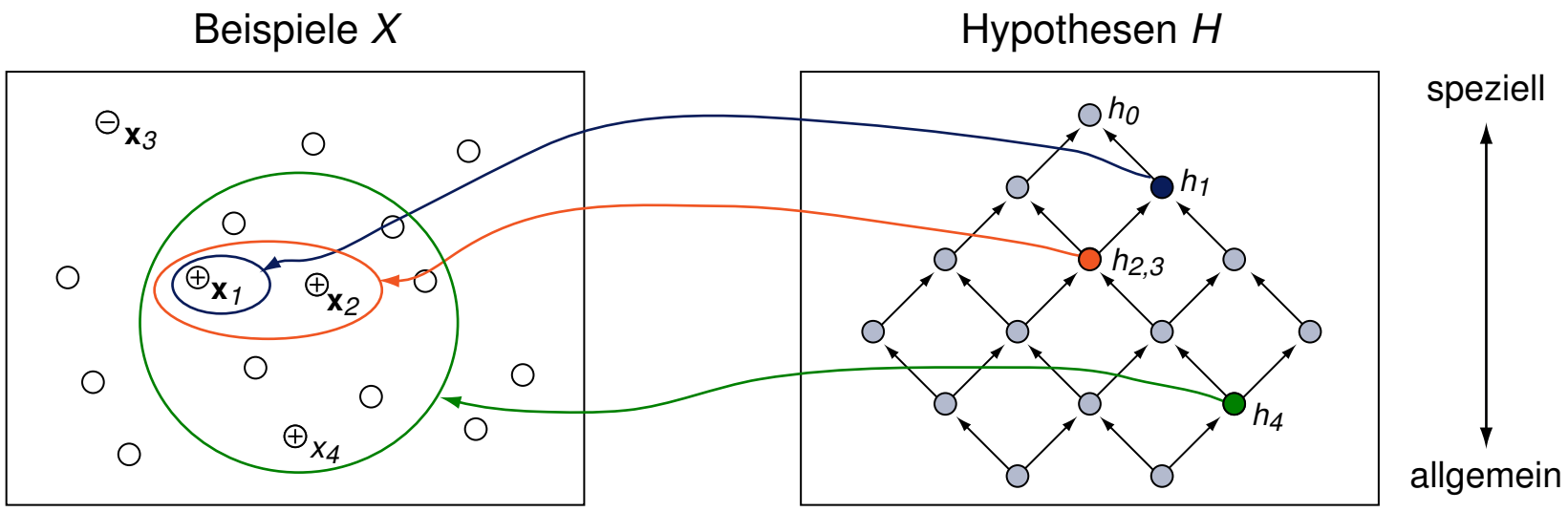
$\mathbf{x}_1 = (\textit{sunny, warm, normal, strong, warm, same})$ $h_1 = \langle \textit{sunny, warm, normal, strong, warm, same} \rangle$

$\mathbf{x}_2 = (\textit{sunny, warm, high, strong, warm, same})$ $h_2 = \langle \textit{sunny, warm, ?, strong, warm, same} \rangle$

$\mathbf{x}_3 = (\textit{rainy, cold, high, strong, warm, change})$ $h_3 = \langle \textit{sunny, warm, ?, strong, warm, same} \rangle$

Konzeptlernen: Suche im Hypothesenraum

Algorithmus Find-S



$h_0 = \langle \perp, \perp, \perp, \perp, \perp, \perp \rangle$

$\mathbf{x}_1 = (\textit{sunny, warm, normal, strong, warm, same})$ $h_1 = \langle \textit{sunny, warm, normal, strong, warm, same} \rangle$

$\mathbf{x}_2 = (\textit{sunny, warm, high, strong, warm, same})$ $h_2 = \langle \textit{sunny, warm, ?, strong, warm, same} \rangle$

$\mathbf{x}_3 = (\textit{rainy, cold, high, strong, warm, change})$ $h_3 = \langle \textit{sunny, warm, ?, strong, warm, same} \rangle$

$\mathbf{x}_4 = (\textit{sunny, warm, high, strong, cool, change})$ $h_4 = \langle \textit{sunny, warm, ?, strong, ?, ?} \rangle$

Konzeptlernen: Suche im Hypothesenraum

Diskussion von Algorithmus Find-S

- Wurde das einzige Konzept gelernt – oder gibt es noch andere?
- Warum sollte die speziellste Hypothese gewählt werden?
- Was passiert, wenn es mehrere maximal spezielle Hypothesen gibt?
- Inkonsistenzen in D können nicht festgestellt werden.

Konzeptlernen: Suche im Version-Space

Definition 4 (Version-Space)

Der Version-Space $V_{H,D}$ eines Hypothesenraums H und einer Trainingsmenge D ist die Menge aller hinsichtlich D konsistenten Hypothesen $h \in H$:

$$V_{H,D} = \{h \mid h \in H \ \wedge \ (\ \forall(\mathbf{x}, c(\mathbf{x})) \in D : \ h(\mathbf{x}) = c(\mathbf{x}) \ ) \ \}$$

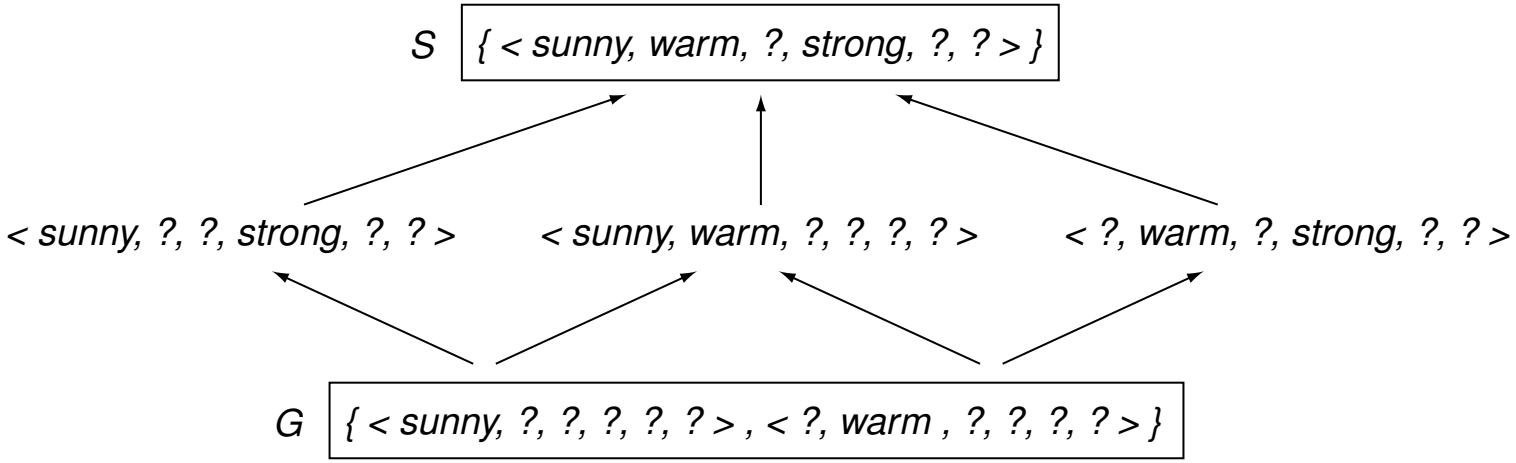
Konzeptlernen: Suche im Version-Space

Definition 4 (Version-Space)

Der Version-Space $V_{H,D}$ eines Hypothesenraums H und einer Trainingsmenge D ist die Menge aller hinsichtlich D konsistenten Hypothesen $h \in H$:

$$V_{H,D} = \{h \mid h \in H \ \wedge \ (\ \forall(\mathbf{x}, c(\mathbf{x})) \in D : \ h(\mathbf{x}) = c(\mathbf{x}) \ ) \ \}$$

Illustration von $V_{H,D}$ für die `Trainingsmenge` D:



Bemerkungen:

- Die Bezeichnung „Version-Space“ drückt aus, dass es sich um die Menge aller konsistenten *Versionen* des gesuchten Zielkonzeptes handelt.
- Ein naiver, nur bei Spielbeispielen praktikabler Ansatz zur Konstruktion des Version-Space besteht in der Erzeugung einer Liste aller Hypothesen und einer anschließenden, sukzessiven Eliminierung derjenigen Hypothesen h, für die $h(\mathbf{x}) \neq c(\mathbf{x})$ gilt. Dieser Ansatz funktioniert nur bei einem endlichen Hypothesenraum.

Konzeptlernen: Suche im Version-Space

Definition 5 (Grenzhypothesen im Version-Space)
Sei H ein Hypothesenraum und sei D eine Trainingsmenge. Dann ist hinsichtlich der $\geq_g$-Relation die Menge der allgemeinsten Hypothesen, G, wie folgt definiert:

$$\{g \mid g \in H \ \wedge \ \textit{consistent}(g, D) \ \wedge \ (\nexists g' : g' \in \ H \ \wedge \ g' >_g \ g \ \wedge \ \textit{consistent}(g', D)) \}$$

Gleichermaßen ist die Menge der speziellsten Hypothesen, S, wie folgt definiert:

$$\{s \mid s \in H \ \wedge \ \textit{consistent}(s, D) \ \wedge \ (\nexists s' : s' \in \ H \ \wedge \ s >_g \ s' \ \wedge \ \textit{consistent}(s', D)) \}$$

Konzeptlernen: Suche im Version-Space

Definition 5 (Grenzhypothesen im Version-Space)

Sei H ein Hypothesenraum und sei D eine Trainingsmenge. Dann ist hinsichtlich der $\geq_g$-Relation die Menge der allgemeinsten Hypothesen, G, wie folgt definiert:

$$\{g \mid g \in H \ \wedge \ \textit{consistent}(g, D) \ \wedge \ (\nexists g' : g' \in \ H \ \wedge \ g' >_g \ g \ \wedge \ \textit{consistent}(g', D)) \}$$

Gleichermaßen ist die Menge der speziellsten Hypothesen, S, wie folgt definiert:

$$\{s \mid s \in H \ \wedge \ \textit{consistent}(s, D) \ \wedge \ (\nexists s' : s' \in \ H \ \wedge \ s >_g \ s' \ \wedge \ \textit{consistent}(s', D)) \}$$

Satz 1 (Version-Space-Repräsentation)

Sei X eine Menge von Beispielen und H eine Menge zweiwertiger Funktionen definiert über X. Weiterhin sei $c : X \rightarrow \{0, 1\}$ ein Zielkonzept und D eine Trainingsmenge mit Beispielen der Form $(\mathbf{x}, c(\mathbf{x}))$. Dann liegt jedes Element des Version-Space $V_{H,D}$ hinsichtlich der $\geq_g$-Relation zwischen zwei Grenzhypothesen:

$$V_{H,D} = \{h \mid h \in H \ \wedge \ (\exists g \in G \ \exists s \in S : g \geq_g h \geq_g s) \}$$

Konzeptlernen: Suche im Version-Space

Candidate-Elimination-Algorithmus

1. $G = \{g_0\}$ // g_0 is the maximally general hypothesis in H.
   $S = \{s_0\}$ // s_0 is the maximally specific hypothesis in H.
2. **FOREACH** $(\mathbf{x}, c(\mathbf{x})) \in D$ **DO**
   IF $c(\mathbf{x}) = 1$ **THEN** // $\mathbf{x}$ is a positive example.
   FOREACH $g \in G$ **DO IF** $g(\mathbf{x}) \neq 1$ **THEN** $G = G \setminus \{g\}$ **ENDDO**
   FOREACH $s \in S$ **DO**
   IF $s(\mathbf{x}) \neq 1$ **THEN**
   $S = S \setminus \{s\}$, $S^+ = \mathit{min\_generalizations}(s)$
   FOREACH $s \in S^+$ **DO IF** $s(\mathbf{x}) = 1 \wedge (\exists g \in G : g \geq_g s)$ **THEN** $S = S \cup \{s\}$ **ENDDO**
   FOREACH $s \in S$ **DO IF** $(\exists s' \in S : s' \neq s \wedge s' \geq_g s)$ **THEN** $S = S \setminus \{s'\}$ **ENDDO**
   ENDDO
   ELSE // $\mathbf{x}$ is a negative example.
   ENDIF
   ENDDO
3. *return*(G, S)

Konzeptlernen: Suche im Version-Space

Candidate-Elimination-Algorithmus

1. $G = \{g_0\}$ // g_0 is the maximally general hypothesis in H.
   $S = \{s_0\}$ // s_0 is the maximally specific hypothesis in H.
2. **FOREACH** $(\mathbf{x}, c(\mathbf{x})) \in D$ **DO**
   IF $c(\mathbf{x}) = 1$ **THEN** // $\mathbf{x}$ is a positive example.
   FOREACH $g \in G$ **DO IF** $g(\mathbf{x}) \neq 1$ **THEN** $G = G \setminus \{g\}$ **ENDDO**
   FOREACH $s \in S$ **DO**
   IF $s(\mathbf{x}) \neq 1$ **THEN**
   $S = S \setminus \{s\}$, $S^+ = \mathit{min\_generalizations}(s)$
   FOREACH $s \in S^+$ **DO IF** $s(\mathbf{x}) = 1 \wedge (\exists g \in G : g \geq_g s)$ **THEN** $S = S \cup \{s\}$ **ENDDO**
   FOREACH $s \in S$ **DO IF** $(\exists s' \in S : s' \neq s \wedge s' \geq_g s)$ **THEN** $S = S \setminus \{s'\}$ **ENDDO**
   ENDDO
   ELSE // $\mathbf{x}$ is a negative example.
   FOREACH $s \in S$ **DO IF** $s(\mathbf{x}) \neq 0$ **THEN** $S = S \setminus \{s\}$ **ENDDO**
   FOREACH $g \in G$ **DO**
   IF $g(\mathbf{x}) \neq 0$ **THEN**
   $G = G \setminus \{g\}$, $G^- = \mathit{min\_specializations}(g)$
   FOREACH $g \in G^-$ **DO IF** $g(\mathbf{x}) = 0 \wedge (\exists s \in S : g \geq_g s)$ **THEN** $G = G \cup \{g\}$ **ENDDO**
   FOREACH $g \in G$ **DO IF** $(\exists g' \in G : g' \neq g \wedge g \geq_g g')$ **THEN** $G = G \setminus \{g'\}$ **ENDDO**
   ENDDO
   ENDIF
   ENDDO
3. *return*(G, S)

Bemerkungen:

Natürlichsprachliche Formulierung der äußeren `FOREACH`-Schleife [vgl. Mitchell 1997] :

- If $\mathbf{x}$ is a positive example
  - Remove from G any hypothesis that is not consistent with $\mathbf{x}$
  - For each hypothesis s in S that is not consistent with $\mathbf{x}$
    - Remove s from S
    - Add to S all minimal generalizations h of s such that
      1. h is consistent with $\mathbf{x}$ and
      2. some member of G is more general than h
    - Remove from S any hypothesis that is less specific than another hypothesis in S
- If $\mathbf{x}$ is a negative example
  - Remove from S any hypothesis that is not consistent with $\mathbf{x}$
  - For each hypothesis g in G that is not consistent with $\mathbf{x}$
    - Remove g from G
    - Add to G all minimal specializations h of g such that
      1. h is consistent with $\mathbf{x}$ and
      2. some member of S is more specific than h
    - Remove from G any hypothesis that is less general than another hypothesis in G

Konzeptlernen: Suche im Version-Space

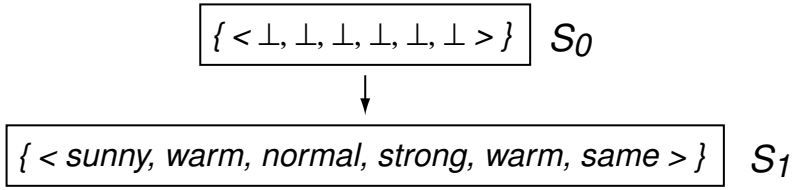
Candidate-Elimination-Algorithmus

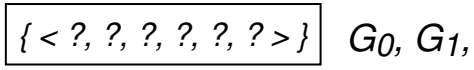
$\boxed{\{ < \bot, \bot, \bot, \bot, \bot, \bot > \}}$ S_0

$\boxed{\{ < ?, ?, ?, ?, ?, ? > \}}$ G_0,

Konzeptlernen: Suche im Version-Space

Candidate-Elimination-Algorithmus

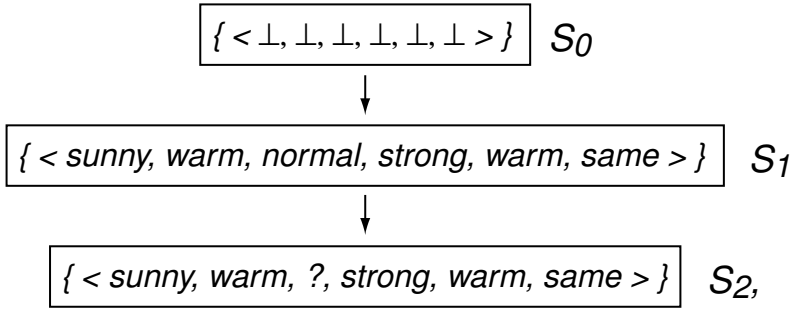




$\mathbf{x}_1 = (\textit{sunny, warm, normal, strong, warm, same})$ $\quad\quad$ $\textit{EnjoySport}(\mathbf{x}_1) = 1$

Konzeptlernen: Suche im Version-Space

Candidate-Elimination-Algorithmus



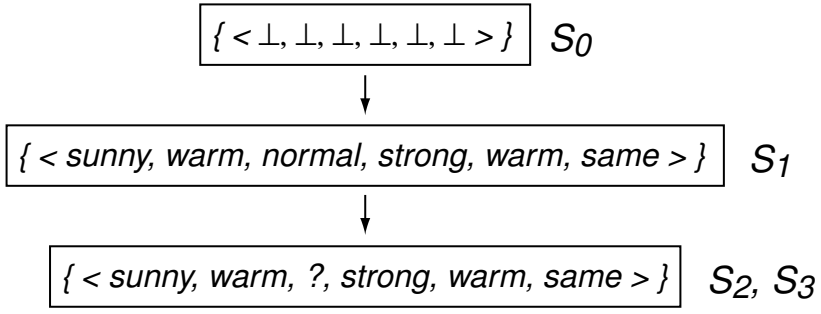
{ < ?, ?, ?, ?, ?, ? > } G_0, G_1, G_2

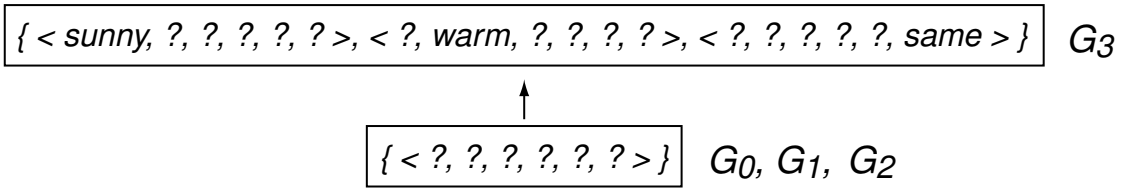
$\mathbf{x}_1 = (\textit{sunny, warm, normal, strong, warm, same})$ $\quad$ $\textit{EnjoySport}(\mathbf{x}_1) = 1$

$\mathbf{x}_2 = (\textit{sunny, warm, high, strong, warm, same})$ $\quad$ $\textit{EnjoySport}(\mathbf{x}_2) = 1$

Konzeptlernen: Suche im Version-Space

Candidate-Elimination-Algorithmus





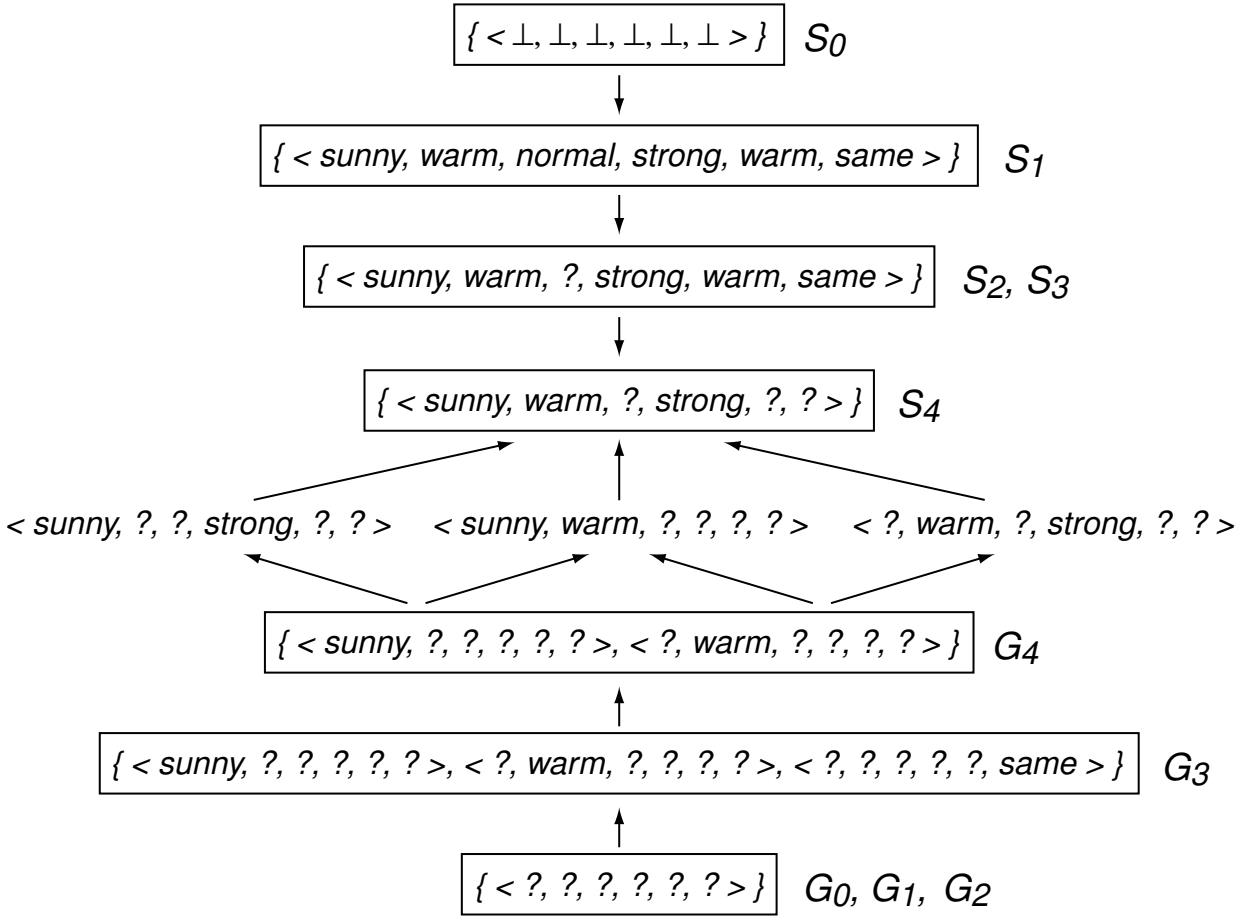
$\mathbf{x}_1 =$ (*sunny, warm, normal, strong, warm, same*) — *EnjoySport*$(\mathbf{x}_1) = 1$

$\mathbf{x}_2 =$ (*sunny, warm, high, strong, warm, same*) — *EnjoySport*$(\mathbf{x}_2) = 1$

$\mathbf{x}_3 =$ (*rainy, cold, high, strong, warm, change*) — *EnjoySport*$(\mathbf{x}_3) = 0$

Konzeptlernen: Suche im Version-Space

Candidate-Elimination-Algorithmus



$\mathbf{x}_1 = (\textit{sunny, warm, normal, strong, warm, same})$ $\quad$ $\textit{EnjoySport}(\mathbf{x}_1) = 1$

$\mathbf{x}_2 = (\textit{sunny, warm, high, strong, warm, same})$ $\quad$ $\textit{EnjoySport}(\mathbf{x}_2) = 1$

$\mathbf{x}_3 = (\textit{rainy, cold, high, strong, warm, change})$ $\quad$ $\textit{EnjoySport}(\mathbf{x}_3) = 0$

$\mathbf{x}_4 = (\textit{sunny, warm, high, strong, cool, change})$ $\quad$ $\textit{EnjoySport}(\mathbf{x}_4) = 1$

Bemerkungen:

- Von der Idee her beschränkt (zunächst) eine speziellste Hypothese $s \in S$ die positiven Beispiele. Deshalb muss für jedes positive Beispiel, das nicht mit s konsistent ist, s relaxiert werden. Umgekehrt toleriert (zunächst) eine allgemeinste Hypothese $g \in G$ die negativen Beispiele. Deshalb muss für jedes negative Beispiel, das nicht mit g konsistent ist, g restriktiver gemacht werden.

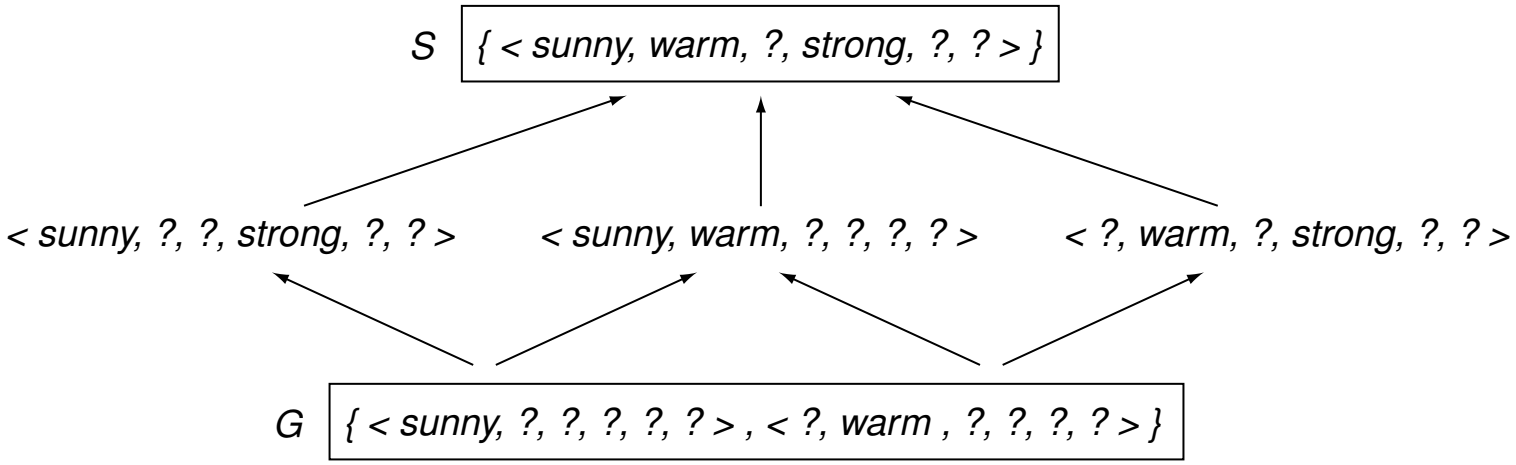
Konzeptlernen: Suche im Version-Space

Diskussion des Candidate-Elimination-Algorithmus

- Konvergiert der Candidate-Elimination-Algorithmus gegen die korrekte Hypothese?
- Wann entsteht ein leerer Version-Space?
- Macht es Sinn, dem Lernalgorithmus bestimmte Trainingsbeispiele vorzusetzen? Stichwort: *Active Learning*
- Was sind teilweise gelernte Konzepte und wie kann man sie benutzen?
- Der hier vorgestellte Version-Space ist „voreingenommen“ *(biased)*. Was bedeutet das?

Konzeptlernen: Suche im Version-Space

Auswahl von Trainingsbeispielen

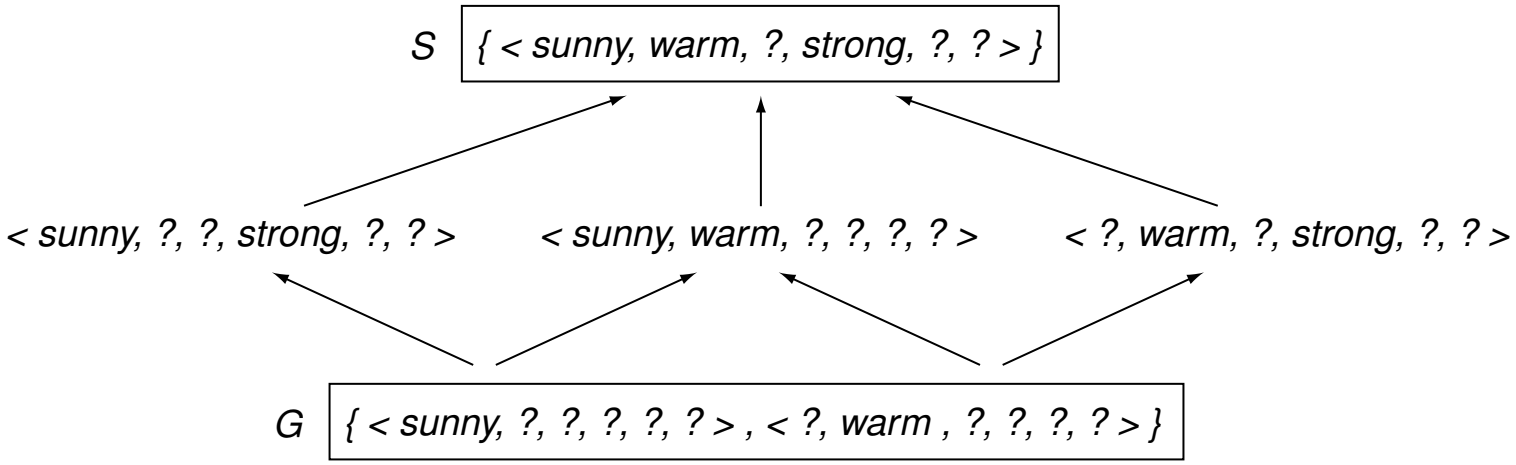


Neues Beispiel:

$\mathbf{x}_c = (\textit{sunny, warm, normal, light, warm, same})$

Konzeptlernen: Suche im Version-Space

Auswahl von Trainingsbeispielen



Neues Beispiel:

$\mathbf{x}_c = (\textit{sunny, warm, normal, light, warm, same})$

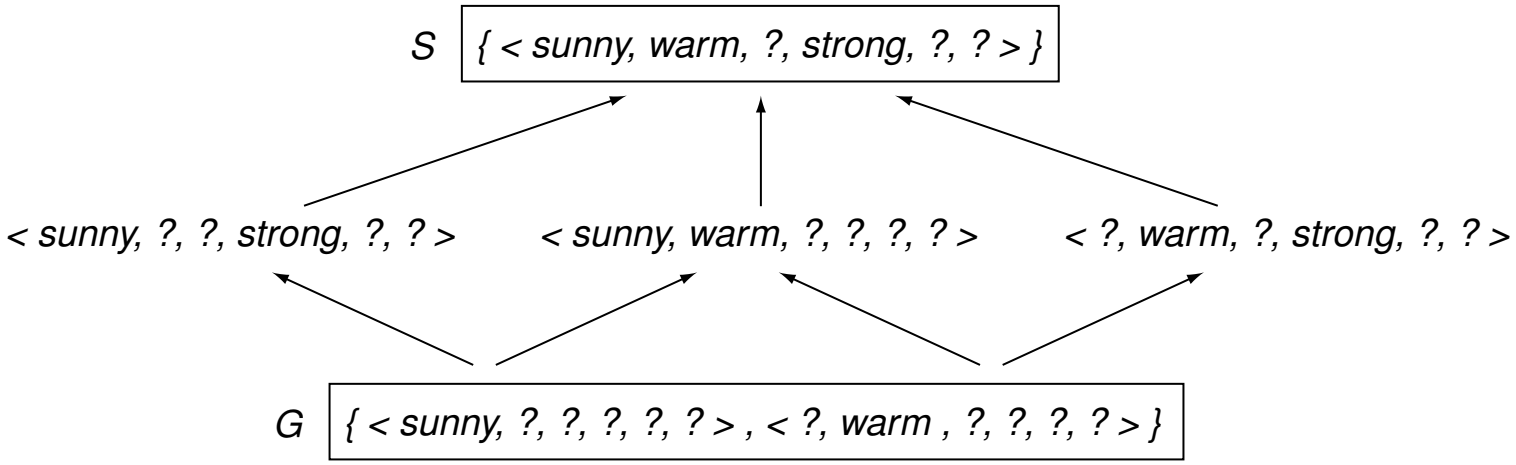
Beobachtung:

Unabhängig davon, welchen Wert $c(\mathbf{x}_c)$ hat, ist $(\mathbf{x}_c, c(\mathbf{x}_c))$ immer konsistent bzgl. drei der sechs Hypothesen. Folglich gilt:

- Mit *EnjoySport*$(\mathbf{x}_c) = 1$ kann die Menge S weiter generalisiert werden.
- Mit *EnjoySport*$(\mathbf{x}_c) = 0$ kann die Menge G weiter spezialisiert werden.

Konzeptlernen: Suche im Version-Space

Teilweise gelernte Konzepte

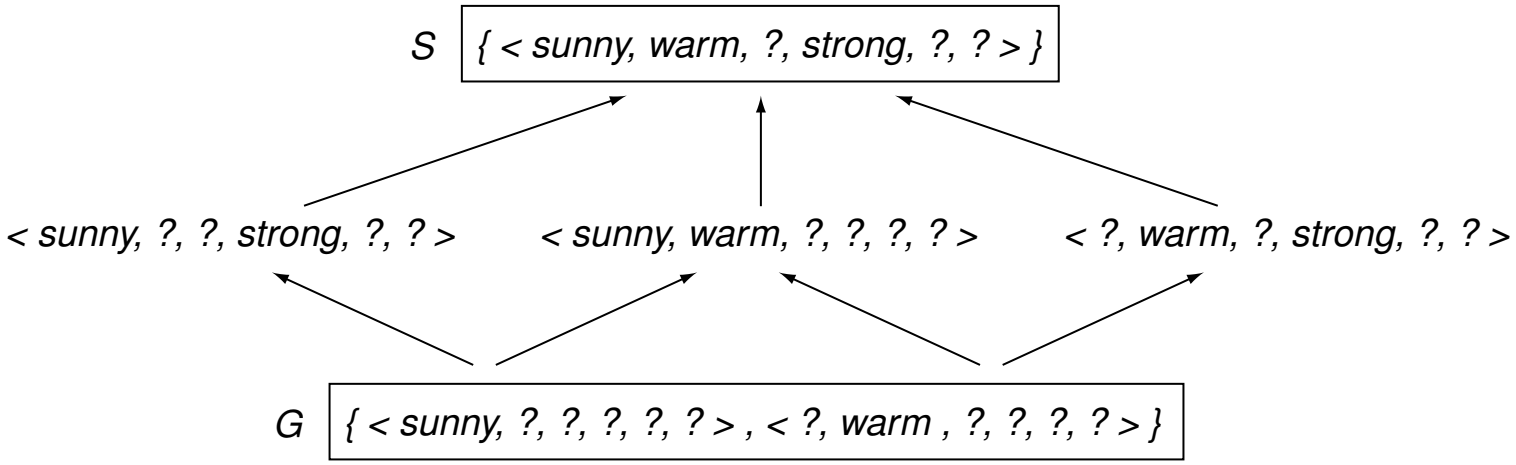


Zu klassifizierende Beispiele mit unbekanntem Zielkonzept:

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(a)	sunny	warm	normal	strong	cool	change	
(b)	rainy	cold	normal	light	warm	same	
(c)	sunny	warm	normal	light	warm	same	
(d)	sunny	cold	normal	strong	warm	same	

Konzeptlernen: Suche im Version-Space

Teilweise gelernte Konzepte

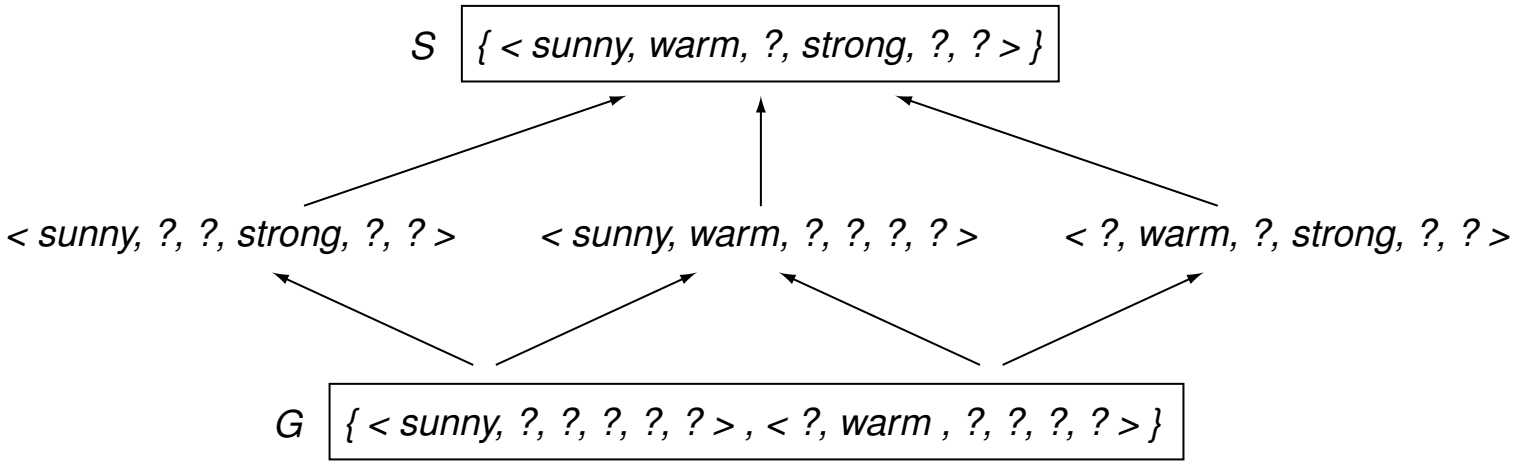


Zu klassifizierende Beispiele mit unbekanntem Zielkonzept:

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(a)	sunny	warm	normal	strong	cool	change	6+ : 0–
(b)	rainy	cold	normal	light	warm	same	
(c)	sunny	warm	normal	light	warm	same	
(d)	sunny	cold	normal	strong	warm	same	

Konzeptlernen: Suche im Version-Space

Teilweise gelernte Konzepte

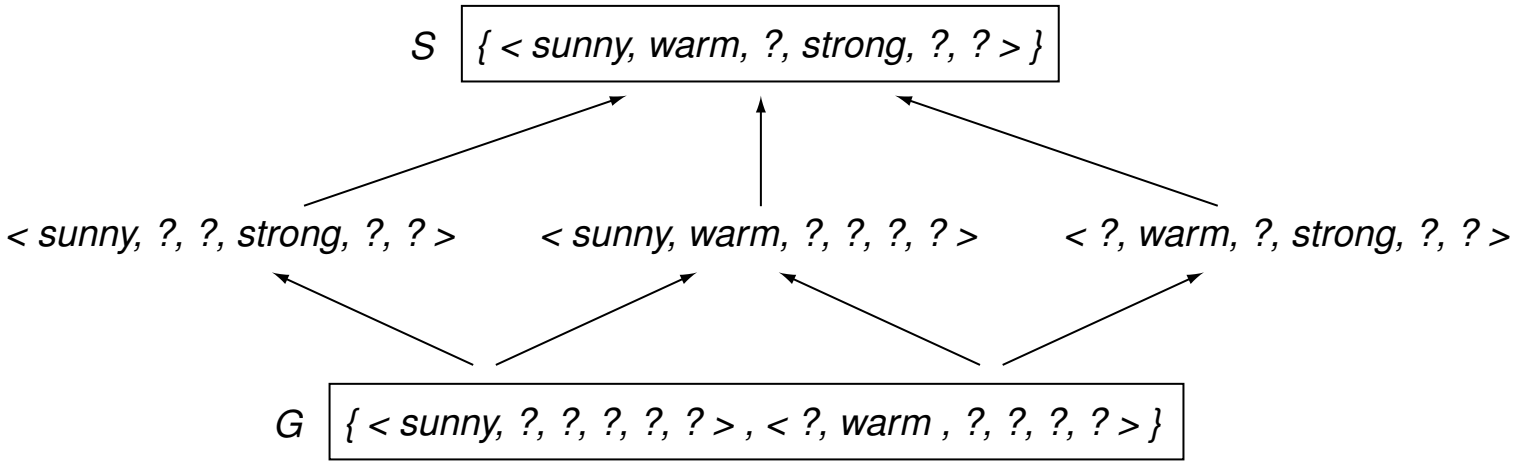


Zu klassifizierende Beispiele mit unbekanntem Zielkonzept:

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(a)	sunny	warm	normal	strong	cool	change	6+ : 0–
(b)	rainy	cold	normal	light	warm	same	0+ : 6–
(c)	sunny	warm	normal	light	warm	same	
(d)	sunny	cold	normal	strong	warm	same	

Konzeptlernen: Suche im Version-Space

Teilweise gelernte Konzepte

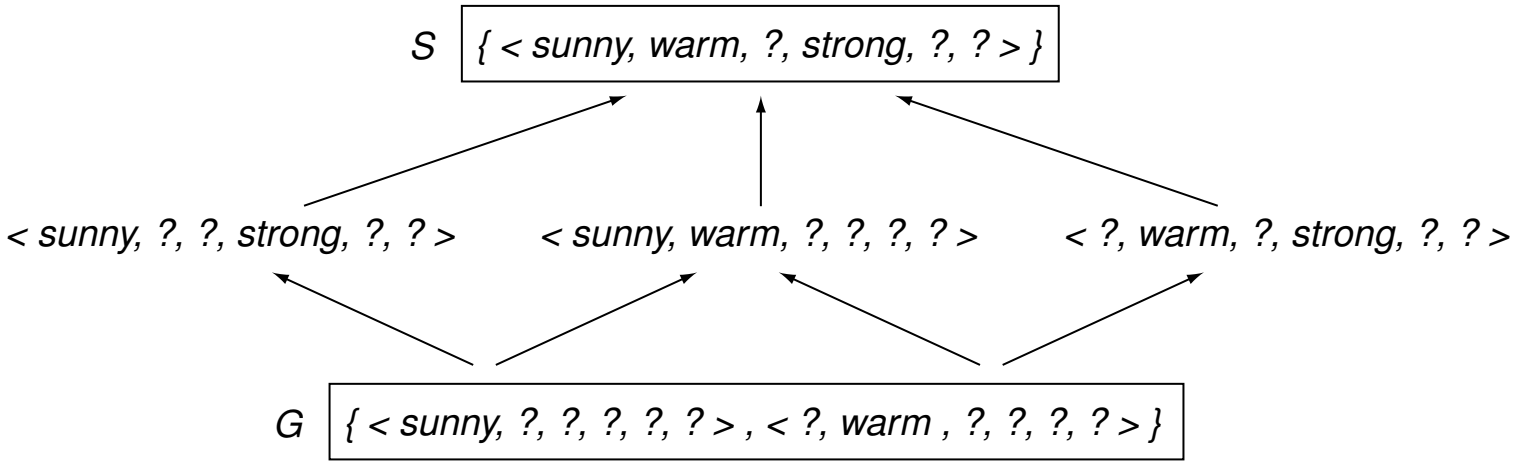


Zu klassifizierende Beispiele mit unbekanntem Zielkonzept:

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(a)	sunny	warm	normal	strong	cool	change	6+ : 0–
(b)	rainy	cold	normal	light	warm	same	0+ : 6–
(c)	sunny	warm	normal	light	warm	same	3+ : 3–
(d)	sunny	cold	normal	strong	warm	same	

Konzeptlernen: Suche im Version-Space

Teilweise gelernte Konzepte



Zu klassifizierende Beispiele mit unbekanntem Zielkonzept:

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(a)	sunny	warm	normal	strong	cool	change	6+ : 0–
(b)	rainy	cold	normal	light	warm	same	0+ : 6–
(c)	sunny	warm	normal	light	warm	same	3+ : 3–
(d)	sunny	cold	normal	strong	warm	same	2+ : 4–

Konzeptlernen: Suche im Version-Space

Inductive Bias

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(e)	sunny	warm	normal	strong	cool	change	yes
(f)	sunny	warm	normal	light	warm	same	yes

➜ $S = \{ \langle \textit{sunny, warm, normal, ?, ?, ?} \rangle \}$

Konzeptlernen: Suche im Version-Space

Inductive Bias

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(e)	sunny	warm	normal	strong	cool	change	yes
(f)	sunny	warm	normal	light	warm	same	yes

$$\Rightarrow \quad S = \{ \langle \textit{sunny, warm, normal, ?, ?, ?} \rangle \}$$

Insbesondere gehen wir davon aus, das folgende Beispiel klassifizieren zu können:

$$\mathbf{x} = (\textit{sunny, warm, normal, strong, warm, same})$$

Was passierte, wenn $\mathbf{x}$ ein negatives Beispiel wäre?

Beobachtung:

Der Lernalgorithmus trifft a-Priori-Annahmen hinsichtlich des Zielkonzepts.

Welche?

Konzeptlernen: Suche im Version-Space

Inductive Bias (Fortsetzung)

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(g)	sunny	warm	normal	strong	cool	change	yes
(h)	cloudy	warm	normal	strong	cool	change	yes

➜ $S = \{ \langle$ *?, warm, normal, strong, cool, change* $\rangle \}$

Konzeptlernen: Suche im Version-Space

Inductive Bias (Fortsetzung)

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(g)	sunny	warm	normal	strong	cool	change	yes
(h)	cloudy	warm	normal	strong	cool	change	yes

➜ $S = \{ \langle$ *?, warm, normal, strong, cool, change* $\rangle \}$

+

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(i)	rainy	warm	normal	strong	cool	change	no

➜ $S = \{ \}$

Konzeptlernen: Suche im Version-Space

Inductive Bias (Fortsetzung)

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(g)	sunny	warm	normal	strong	cool	change	yes
(h)	cloudy	warm	normal	strong	cool	change	yes

➜ $S = \{ \langle$ *?, warm, normal, strong, cool, change* $\rangle \}$

+

Example	Sky	Temperature	Humidity	Wind	Water	Forecast	EnjoySport
(i)	rainy	warm	normal	strong	cool	change	no

➜ $S = \{ \}$

Beobachtung:

Der Hypothesenraum H müsste so gewählt werden, dass er jedes mögliche Konzept enthält – u. a.: $\langle$ *sunny, ?, ?, ?, ?, ?* $\rangle \vee \langle$ *cloudy, ?, ?, ?, ?, ?* $\rangle$

Konzeptlernen: Suche im Version-Space

Inductive Bias (Fortsetzung)

- Ein Lernalgorithmus, der alle möglichen Hypothesen als gleich wahrscheinlich betrachtet, trifft keine a-Priori-Annahme hinsichtlich des Zielkonzepts.
- Ein Lernalgorithmus ohne a-Priori-Annahmen hat keinen „induktiven Bias".

  “The policy by which an algorithm generalizes from observed training examples to classify unseen instances is its inductive bias. [. . .] Inductive bias is the set of assumptions that, together with the training data, deductively justify the classification by the learner to future instances.”

  [Mitchell 1997, p.63]

Konzeptlernen: Suche im Version-Space

Inductive Bias (Fortsetzung)

- Ein Lernalgorithmus, der alle möglichen Hypothesen als gleich wahrscheinlich betrachtet, trifft keine a-Priori-Annahme hinsichtlich des Zielkonzepts.
- Ein Lernalgorithmus ohne a-Priori-Annahmen hat keinen „induktiven Bias".

  "The policy by which an algorithm generalizes from observed training examples to classify unseen instances is its inductive bias. [. . .] Inductive bias is the set of assumptions that, together with the training data, deductively justify the classification by the learner to future instances."

  [Mitchell 1997, p.63]

- Ein Lernalgorithmus ohne induktiven Bias hat keine Grundlage, um nicht-gesehene Beispiele zu klassifizieren. Er kann *nicht generalisieren*.
- Ein Lernalgorithmus ohne induktiven Bias lernt lediglich auswendig.
- Welcher der beiden Algorithmen Find-S und Candidate-Elimination hat einen stärkeren induktiven Bias?

Kapitel ML: II (Fortsetzung)

II. Grundlagen des Maschinellen Lernens

- Daten
- Datenexploration
- Konzeptlernen: Suche im Hypothesenraum
- Konzeptlernen: Suche im Version-Space
- Performance Measures

Performance Measures

Missklassifikationsrate

Für einen Klassifikator $y : X \to C$ und ein zu lernendes Konzept c bezeichnet *Err*$^*(c, y)$ die wahre Missklassifikationsrate *(true misclassification rate)*.

Für endliches X ist *Err*$^*(c, y) = \dfrac{|\{\mathbf{x} \in X \mid c(\mathbf{x}) \neq y(\mathbf{x})\}|}{|X|}$

Problem: Die Funktion c ist meist unbekannt.

➜ Schätzung von *Err*$^*(c, y)$ durch Auswertung von y auf einer Testmenge.
Da c im Kontext immer fest gewählt ist, verwenden wir *Err*$^*(y)$ statt *Err*$^*(c, y)$.

Performance Measures

Missklassifikationsrate

Wahrscheinlichkeitstheoretische Fundierung:

- P Wahrscheinlichkeitsmaß auf $X \times C$.
- $P(A, j)$ für $A \subseteq X$ und $j \in C$ Wahrscheinlichkeit, dass ein zufällig gezogenes $\mathbf{x} \in X$ in die Menge A fällt und zur Klasse j gehört.
- $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_N, c(\mathbf{x}_N))\} \subseteq X \times C$ Menge von aus $X \times C$ nach der Verteilung P unabhängig voneinander gezogenen Beispielen.
- $y : X \to C$ vom Lernalgorithmus auf Basis von D bestimmte Klassifikator.
- $(\mathbf{x}_0, c(\mathbf{x}_0))$ unabhängig von D nach P gezogen, d.h. $P(\mathbf{x}_0 \in A, c(\mathbf{x}_0) = j) = P(A, j)$ und $(\mathbf{x}_0, c(\mathbf{x}_0))$ unabhängig von D.

Missklassifikationsrate: $Err^*(y) = P(c(\mathbf{x}_0) \neq y(\mathbf{x}_0) \mid D)$

Performance Measures

Missklassifikationsrate

Schätzung von $Err^*(y)$ auf Basis einer endlichen Teilmenge T von X:

$$Err(y, T) = \frac{|\{\mathbf{x} \in T \mid c(\mathbf{x}) \neq y(\mathbf{x})\}|}{|T|}$$

Für $\mathbf{x} \in T$ muss $c(\mathbf{x})$ bekannt sein, also $T \subseteq D$.

Performance Measures

Trainingsfehler (Resubstitutionsfehler)

- Trainingsmenge $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$.
- $y : X \to C$ sei der vom Lernalgorithmus auf Basis von D bestimmte Klassifikator.

Trainingsfehler = Missklassifikationsrate auf der Trainingsmenge D_{tr} :

$$\mathit{Err}(y, D) = \frac{|\{(\mathbf{x}, c(\mathbf{x})) \in D \mid c(\mathbf{x}) \neq y(\mathbf{x})\}|}{|D|}$$

Performance Measures

Trainingsfehler (Resubstitutionsfehler)

- Trainingsmenge $D = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq X \times C$.
- $y : X \to C$ sei der vom Lernalgorithmus auf Basis von D bestimmte Klassifikator.

Trainingsfehler = Missklassifikationsrate auf der Trainingsmenge D_{tr} :

$$\mathit{Err}(y, D) = \frac{|\{(\mathbf{x}, c(\mathbf{x})) \in D \mid c(\mathbf{x}) \neq y(\mathbf{x})\}|}{|D|}$$

Problem:

- Missklassifikationsrate wird für die Menge von Beispielen bestimmt, die schon zum Lernen benutzt wurden.

➜ Auswendiglernen führt zu minimalem Trainingsfehler.

➜ Trainingsfehler ist meist eine zu optimistische Schätzung.

Performance Measures

Holdout-Schätzung

- Trainingsmenge $D_{tr} = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq D \subseteq X \times C$.
- $y : X \to C$ sei auf Basis von D_{tr} gelernte Klassifikator.
- Testmenge $D_{ts} = \{(\mathbf{x}'_1, c(\mathbf{x}'_1)), \ldots, (\mathbf{x}'_{n'}, c(\mathbf{x}'_{n'}))\} \subseteq D$, $D_{ts} \cap D_{tr} = \emptyset$.

Holdout-Schätzung = Missklassifikationsrate auf der Testmenge D_{ts} :

$$\mathit{Err}(y, D_{ts}) = \frac{|\{(\mathbf{x}', c(\mathbf{x}')) \in D_{ts} \mid c(\mathbf{x}') \neq y(\mathbf{x}')\}|}{|D_{ts}|}$$

Performance Measures

Holdout-Schätzung

- Trainingsmenge $D_{tr} = \{(\mathbf{x}_1, c(\mathbf{x}_1)), \ldots, (\mathbf{x}_n, c(\mathbf{x}_n))\} \subseteq D \subseteq X \times C$.
- $y : X \to C$ sei auf Basis von D_{tr} gelernte Klassifikator.
- Testmenge $D_{ts} = \{(\mathbf{x}'_1, c(\mathbf{x}'_1)), \ldots, (\mathbf{x}'_{n'}, c(\mathbf{x}'_{n'}))\} \subseteq D$, $D_{ts} \cap D_{tr} = \emptyset$.

Holdout-Schätzung = Missklassifikationsrate auf der Testmenge D_{ts} :

$$\mathit{Err}(y, D_{ts}) = \frac{|\{(\mathbf{x}', c(\mathbf{x}')) \in D_{ts} \mid c(\mathbf{x}') \neq y(\mathbf{x}')\}|}{|D_{ts}|}$$

Probleme:

- Testdaten und Trainingsdaten müssen unabhängig nach der gleichen Verteilung gezogen sein.
- Testdatenmenge und Trainingsdatenmenge müssen groß sein.

Bemerkungen:

- Die Aufteilung von D geschieht oft im Verhältnis 2:1 in Trainingsmenge D_{tr} und Testmenge D_{ts}.
- Bei der Aufteilung in Trainings- und Testmenge muss darauf geachtet werden, dass die Verteilung erhalten bleibt, z.B. in beiden Mengen die Objektklassen in gleicher relativer Häufigkeit auftreten (Stratifizierung).

Performance Measures

k-fache Kreuzvalidierung (k-fold cross-validation)

Bessere Vorgehensweise bei kleinen Beispielmengen:

- Zerlege D in möglichst gleichgroße, disjunkte Teilmengen $D_1, \ldots, D_k$.
- Für $i = 1, \ldots, k$
  1. wende das Lernverfahren an auf $D \setminus D_i$, bestimme $y_i : X \to C$ und
  2. bestimme $\mathit{Err}(y_i, D_i) = \frac{|\{(\mathbf{x}, c(\mathbf{x}) \in D_i | y_i(\mathbf{x}) \neq c(\mathbf{x})\}|}{|D_i|}$.
- Wende das Lernverfahren an auf D und bestimme $y : X \to C$.

Kreuzvalidierung-Missklassifikationsrate:

$$\mathit{Err}_{cv}(y, D, k) = \frac{1}{k} \sum_{i=1}^{k} \mathit{Err}(y_i, D_i)$$

Bemerkungen:

- Annahme: Für großes k ist $D \setminus D_i$ fast so groß wie D und damit *Err*$^*(y_i)$ nahe bei *Err*$^*(y)$.
- Bei Entscheidungsbaumlernen ist $k = 10$ oft eine gute Wahl.

Performance Measures

Leave-One-Out-Kreuzvalidierung (leave one out cross-validation)

Spezialfall der Kreuzvalidierung mit $k = n$:

- Bestimme die Missklassifikationsrate bei Kreuzvalidierung für die Mengen $D_i = D \setminus \{(\mathbf{x}_i, c(\mathbf{x}_i))\}$ mit $k \in \{1, ..., n\}$.

Performance Measures

Leave-One-Out-Kreuzvalidierung (leave one out cross-validation)

Spezialfall der Kreuzvalidierung mit $k = n$:

- Bestimme die Missklassifikationsrate bei Kreuzvalidierung für die Mengen $D_i = D \setminus \{(\mathbf{x}_i, c(\mathbf{x}_i))\}$ mit $k \in \{1, ..., n\}$.

Probleme:

- Großer Rechenaufwand bei größeren Beispielmengen.
- Einelementige Testmengen sind nicht stratifiziert (nur eine Klasse).
- Fehlerüberschätzungen sind möglich: Lernverfahren „Mehrheitsentscheidung auf Trainingsmenge“ liefert 100% Missklassifikationsrate bei Beispielmenge mit zwei Klassen und gleicher Beispielanzahl für jede Klasse.

Performance Measures

Bootstrapping

- Ausgangspunkt Lernmenge D mit n Beispielen
- Für $i = 1, \ldots, k$
  1. ziehe aus D insgesamt n Beispiele mit Zurücklegen und bilde daraus die Lernmenge D_i,
  2. wende das Lernverfahren an auf D_i, bestimme $y_i : X \to C$ und
  3. bestimme $\mathit{Err}(y_i, D \setminus D_i) = \frac{|\{(\mathbf{x}, c(\mathbf{x})) \in D \setminus D_i | y_i(\mathbf{x}) \neq c(\mathbf{x})\}|}{|D \setminus D_i|}$.
- Wende das Lernverfahren an auf D und bestimme $y : X \to C$.

Bootstrapping-Missklassifikationsrate :

$$\mathit{Err}_{bt}(y, D) = \frac{1}{k} \sum_{i=1}^{k} \mathit{Err}(y_i, D \setminus D_i)$$

Bemerkung:

- Die Wahrscheinlichkeit, dass ein Beispiel mindestens einmal gezogen wird, ist $1 - (1 - 1/n)^n$.
- Falls n groß gilt $1 - (1 - 1/n)^n \approx 1 - 1/e \approx 0.632$.
- In jeder Lernmenge sind etwa 63.2% der Beispiele in D.
- Man kann die Klassifikatoren $y_1, \ldots, y_k$ auch zu einem Ensemble zusammenfassen und durch Mehrheitsentscheid die Klasse eines Beispiels festlegen:

$$y(\mathbf{x}) := \operatorname*{argmax}_{j \in C} |\{i \in \{1, \ldots, k\} \mid y_i(\mathbf{x}) = j\}|$$

  Verbesserungen der Fehlerrate von 20% bis 47% bei Anwendung mit Entscheidungsbäumen gegenüber einfachem Entscheidungsbaumlernen wurden beobachtet.